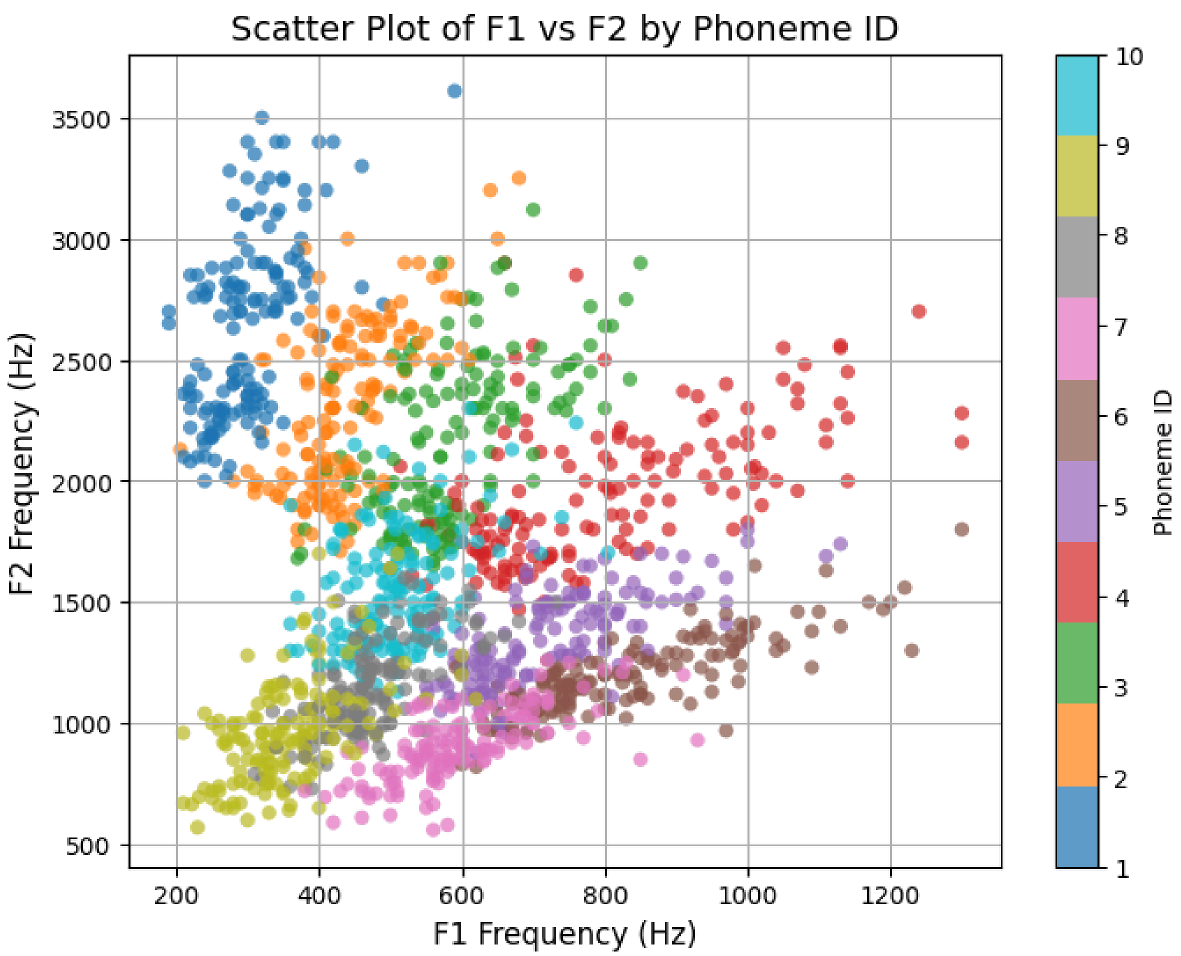
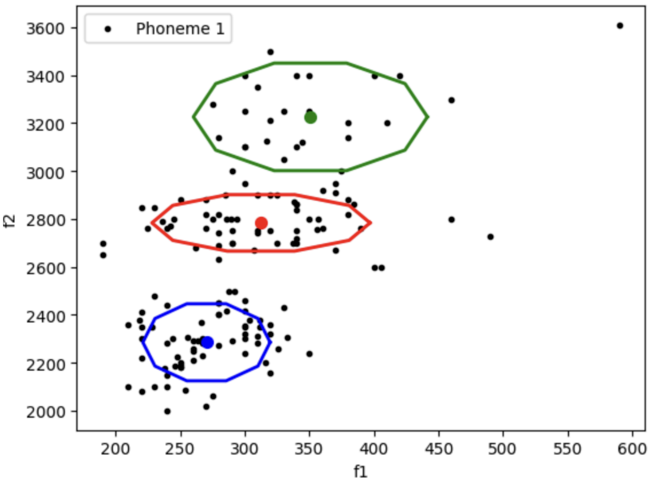
Q1. Produce a plot of $F1$against $F2$. (You should be able to spot some clusters already in this scatter plot.). Comment on the figure and the visible clusters [2 marks]

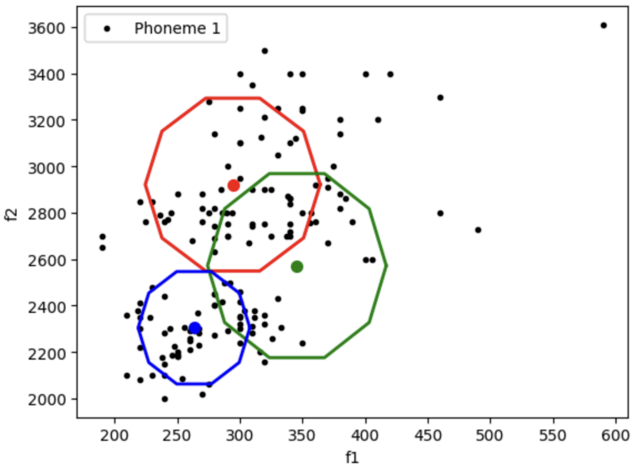


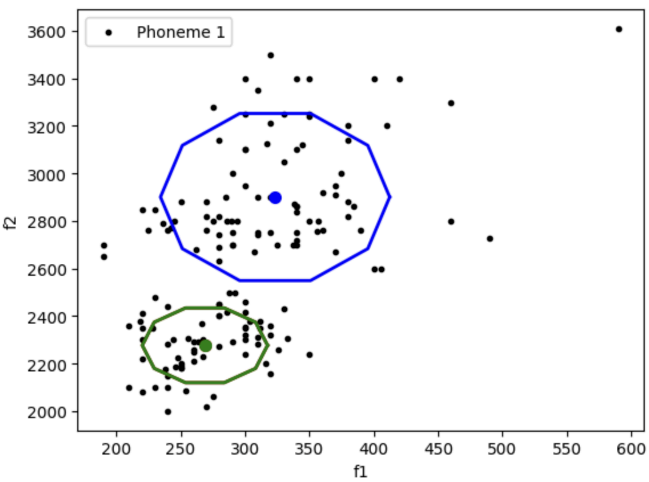
The scatter plot above effectively illustrates the natural grouping of phonemes based on the fundamental frequencies F1 and F2. The graph reveals several distinct clusters, with each cluster corresponding to a specific phoneme ID. These clusters are formed based on the relationship between the F1 and F2 frequencies. Some clusters are more tightly packed, indicating phonemes that are closer in terms of frequency values, while others are more spread out. Some overlap between clusters is visible, suggesting that certain phonemes may share similar frequency features. However, the majority of clusters are well-separated, showing that the F1 and F2 frequencies are good features for distinguishing between phonemes. The clusters' distinctness and separation suggest that further processing, such as classification, could be applied to accurately categorize the phonemes. However we must also consider the overlapping parts of the clusters

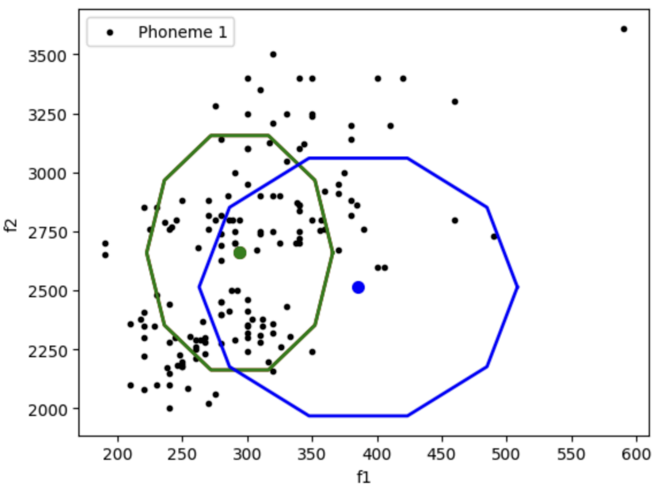
Q2. Run the code multiple times for K=3 , what do you observe? Use figures and the printed MoG parameters to support your arguments [2 mark]

Below are the results from the code after running it twice:









Note that the graphs in which we see 3 clusters are results of the first run and the graphs with 2 clusters are the results of the second run; this means that there is variability in the clustering. The variability in clustering outcomes between runs is due to the random initialization of Gaussian means (μ) and weights (π). In the first run, the initialization led to convergence to three distinct clusters, while in the second run, two clusters merged into one. Running the code multiple times for K=3 results in different clustering outcomes due to the sensitivity of the EM algorithm to initialization

Observations from the First Run (3 Clusters): The clusters are well-separated into three Gaussian components. The red, green, and blue ellipses (representing covariance matrices) cover distinct regions of the data points.The blue cluster at the bottom is tighter and more compact, with smaller variance.The green and red clusters are more spread out, indicating larger variances. Most data points seem clearly assigned to one of the three clusters with minimal overlap between them.

Observations from the Second Run: Despite setting K=3, the result shows only two distinct Gaussian components in the final clustering. This suggests that one cluster (likely from the initialization) merged with another during the EM optimization process. The blue and green ellipses are now larger and cover overlapping regions.This overlap indicates that the clusters are less distinct compared to the first run.

Q5. Use the 2 MoGs (K=3) learnt in tasks 2 & 3 to build a classifier to discriminate between phonemes 1 and 2, and explain the process in the report [4 marks]

To classify data points between phoneme 1 and phoneme 2, we utilized the Maximum Likelihood (ML) criterion based on the Gaussian Mixture Models (GMMs) trained in previous tasks with K=3. The dataset containing data points for both phonemes was combined, and the true phoneme labels were retained for evaluation. Pretrained GMM parameters (μ,Σ,π) for each phoneme were loaded from saved files, with each GMM representing the probability distribution of its respective phoneme. For every data point, the likelihood of belonging to each phoneme was calculated: p(x∣θ1) for phoneme 1 and p(x∣θ 2) for phoneme 2. These likelihoods were computed by summing the weighted contributions from each Gaussian component in the respective GMMs. Each point was classified as belonging to the phoneme with the higher likelihood. The classification accuracy was determined by comparing the predicted phoneme labels to the true labels, yielding an accuracy of 22.04% and a miss-classification error of 77.96%. This low accuracy indicates that the GMMs were not able to effectively model the phoneme distributions for K=3. The poor performance is likely due to overlapping regions between the phoneme distributions in the feature space (F1 and F2), insufficient components (K=3) to represent the complex distributions, or limitations of the diagonal covariance assumption in the GMMs. Future improvements could involve increasing the number of GMM components (e.g., K=6) or using additional features to provide better separation between phonemes. Additionally, using full covariance matrices or testing multiple GMM initializations could improve the model's capacity to capture complex feature correlations.

Q6. Repeat for K=6 and compare the results in terms of accuracy. [2 mark]

For k=3:

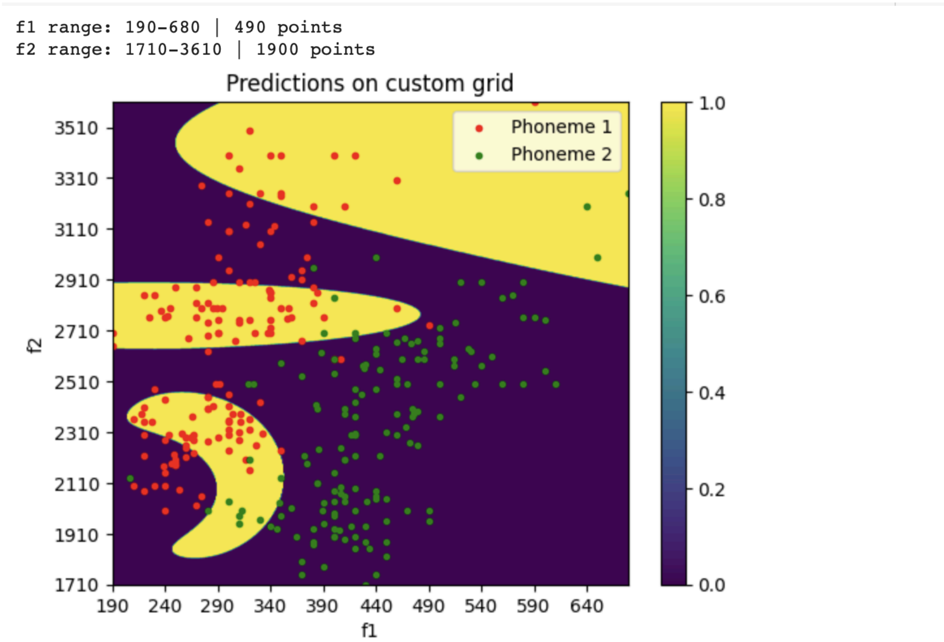
- **`Accuracy: 0.2204`**
- **`Miss-classification Error: 0.7796`**

For k=6:

- **`Accuracy: 0.7007`**
- **`Miss-classification Error: 0.2993`**

When comparing the classifier's performance for K=3 and K=6, the accuracy improved significantly when increasing the number of Gaussian components. With K=3, the model achieved an accuracy of 22.04% and a miss-classification error of 77.96%, which is close to random guessing and indicates that the GMMs with K=3 struggled to model the phoneme distributions effectively. However, with K=6, the accuracy increased to 70.07%, with a miss-classification error of 29.93%, demonstrating an improvement in the classifier's ability to distinguish between phoneme 1 and phoneme 2. The increase in K allowed the GMMs to better capture the complexity of the phoneme distributions by fitting more Gaussian components, making the models more expressive. This flexibility helped the models account for variability in the data, reducing overlap between the distributions for phoneme 1 and phoneme 2. The classifier was able to better separate the two phonemes, leading to the observed improvement in accuracy.However, while the accuracy improved with K=6, it still indicates that the feature space (F1, F2) has limitations in fully separating the two phoneme classes. Some overlap likely remains, preventing the classifier from achieving higher accuracy. Additionally, increasing K adds more parameters to the model, which could lead to overfitting if the dataset were smaller or noisier. In conclusion, increasing K from 3 to 6 significantly improved classification accuracy, demonstrating that K=6 better represents the complexity of the phoneme distributions. Future steps could involve further increasing K, exploring additional features, or using regularization to balance model complexity and generalization.

Q7. Display a "classification matrix" assigning labels to a grid of all combinations of the $F1$ and $F2$ features for the K=3 classifiers from above. Next, repeat this step for K=6 and compare the two. [3 marks]



The above graph shows the classification matrix for K=3, where the yellow regions correspond to areas classified as phoneme 1, and the dark purple regions correspond to phoneme 2. The boundaries between the two regions are determined by the Maximum Likelihood (ML) criterion based on the probabilities generated by the GMMs. These probabilities are calculated using the three Gaussian components for each phoneme. The decision boundaries are smooth and capture the general distinctions between the two phonemes.

However, as seen in the graph, there are some overlaps between the yellow and purple regions, particularly in areas where the phoneme distributions are not well-separated. These overlaps lead to misclassified points, where the model assigns a phoneme label incorrectly due to the similarity of the feature values (F1 and F2) between the two phonemes. This indicates the model's limitations with K=3, as it struggles to fully capture the complexity of the phoneme distributions.

When K=6 is used, the decision boundaries become more detailed and intricate. This is because the GMMs for each phoneme now have six Gaussian components, allowing them to model more complex and nuanced patterns in the data. With K=6, the classifier can better adapt to the shapes of the phoneme data, potentially reducing misclassification in areas where K=3 was too simple. However, the increased flexibility can also lead to overfitting, especially in regions with sparse data, where the model might create overly complex boundaries that do not generalize well to new data.

Below is a comparison of k=3 and k=6:

For K=3:

- Simpler decision boundaries, reflecting the main differences between the two phonemes.
- Generalized solution that avoids overfitting but may fail to capture finer details of the distributions.
- Higher misclassification in overlapping areas due to insufficient modeling complexity.

For K=6:

- More detailed boundaries, better matching the shapes of the phoneme distributions.
- Reduced misclassification in well-represented areas, as the model captures more specific patterns.
- Risk of overfitting in regions with limited data, leading to overly complex decision boundaries.

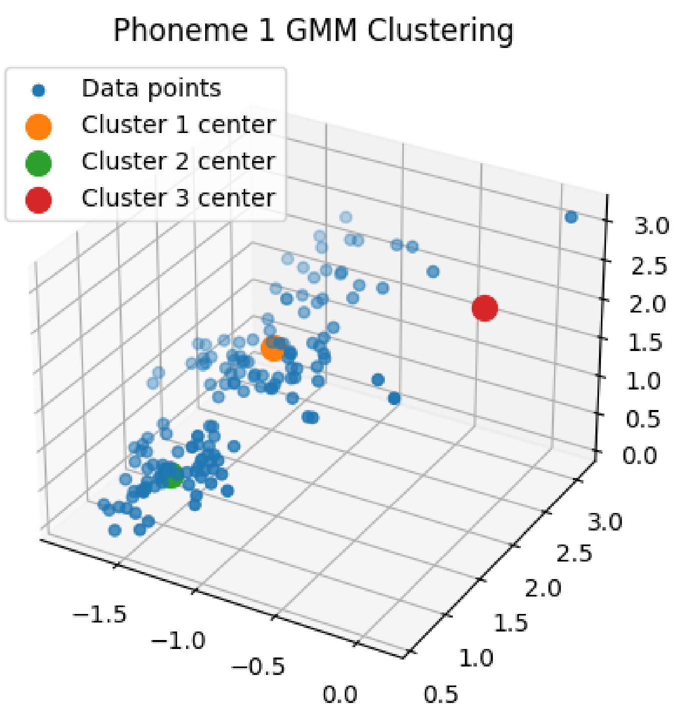
The classification matrix for K=3 provides a simpler and more general representation of the data, while K=6 results in more detailed decision boundaries that align more closely with the data. While K=6 improves accuracy overall, as seen in the earlier results, it may introduce unnecessary complexity in sparse regions. This highlights the trade-off between simplicity and flexibility when choosing the number of Gaussian components in the GMM.

Q8. Try to fit a MoG model to the new data. What is the problem that you observe? Explain why it occurs [2 marks]

The problem observed during the Expectation-Maximization (EM) algorithm occurs because of numerical instability during the E-step. Specifically, the determinant of the covariance matrix (s_i_det) becomes very small or zero due to the dataset's structure, leading to a division by zero or extremely large values during the calculation of Gaussian probabilities. This results in "infinity" (inf) or "not-a-number" (NaN) values in the responsibility matrix (Z), causing the algorithm to fail. The cause of this issue is the dataset X = [F1, F2, F1+F2], where the third feature (F1+F2) is a linear combination of the first two features (F1 and F2). This linear dependence makes the covariance matrix singular or nearly singular, as the data points effectively lie on a 2D plane in a 3D space. Consequently, the determinant of the covariance matrix becomes zero or close to zero, and the inverse of the matrix cannot be computed reliably which leads to numerical errors.In addition, the exponential function used in the Gaussian probability calculation amplifies these issues, as even small numerical errors can result in excessively large or small values. These instabilities prevent the EM algorithm from converging properly, leading to failure in fitting the MoG model.

Q9. Suggest ways of overcoming the singularity problem and implement one of them. Show any training outputs in the report and discuss. [3 marks]

To overcome the singularity problem in the Expectation-Maximization (EM) algorithm for the Gaussian Mixture Model (GMM), we implemented the regularization term approach. During the M-step, when updating the covariance matrix s[i] for each Gaussian component, we added a small constant (epsilon) to its diagonal. This ensures that the covariance matrix remains invertible and avoids issues caused by near-singular or singular matrices. Adding epsilon introduces a slight bias, guaranteeing that the matrix has a minimum variance along each dimension. This stabilizes the inversion of the covariance matrix during the E-step, where responsibilities (Z) are computed, and prevents numerical instabilities such as divide-by-zero errors or excessively large values. The GMM was trained using a dataset with features F1, F2, and F1 + F2, filtered to include only data points corresponding to phoneme 1 (p_id = 1). The visualization shows the data points and the cluster centers of the three GMM components. The regularization successfully resolved the singularity problem, and the model converged without any numerical errors during training.



Above is the output 3D scatter plot of the proposed solution. It provides a clear depiction of the data points (blue) and the centers of the three GMM clusters (orange, green, and red). The placement of the cluster centers demonstrates that the GMM has effectively grouped the data into distinct regions of the feature space based on the Gaussian components. The orange and green cluster centers are close to high-density areas of the data, suggesting that the GMM successfully captured the structure of these clusters. The red cluster center appears farther from the densest areas, indicating that this component may have a broader variance or may account for less prominent patterns in the data.